

# LabS 实验报告

---

## 一、阅读代码

---

该实验代码包括内存模块，寄存器模块，模拟器模块，以及主函数。

模拟的大致流程是打开一个二进制代码，对其逐行读入，存入X3000开头的内存模块中，用PC读取其中的指令，对每条指令取其前四位操作码跳转到对应的执行函数，进行相应的操作。

## 二、配置环境

---

使用wsl，进行sudo apt update, sudo apt upgrade, sudo apt install libboost操作实现相关环境的配置。

## 三、代码填写

---

### 1.存入内存

```
void memory_tp::ReadMemoryFromFile(std::string filename, int
beginning_address) {
    int16_t result;
    std::ifstream input_file;
    input_file.open(filename);
    input_file.is_open();//打开文件
    //读一行后用代码转换函数得到代码，再存入内存
    std::string code_line;
    while (getline(input_file, code_line))
    {
        std::cout << code_line << std::endl;
        result = TranslateInstruction(code_line);
        memory[beginning_address] = result;
        beginning_address ++;
    }
}
```

使用getline 函数读入每一行，用ReadMemoryFromFile函数转换指令，存入内存。

### 2.位拓展函数

```

template <typename T, unsigned B>
inline T SignExtend(const T x) {
    // Extend the number
    // TO BE DONE
    short int judge = (0x1 << (B - 1)); //判断最高位
    short int temp = ~(0x1 << B) - 1; //高位全变为1，低位为0;
    if(judge & x)
        return x | temp; //最高位为1
    else
        return x; //最高位为0
}

```

对传入的数的最高位进行分析，若为1将高位均置1，若为0，高位置0；

### 3.状态符函数

```

void virtual_machine_tp::UpdateCondRegister(int regname) {
    // Update the condition register
    // TO BE DONE
    if(reg[regname] == 0)
        reg[R_COND] = 0b010;
    else if(reg[regname] < 0)
        reg[R_COND] = 0b100;
    else reg[R_COND] = 0b001;
}

```

### 4.AND函数等

```

void virtual_machine_tp::VM_AND(int16_t inst) {
    // DONE
    int flag = inst & 0b100000;
    int dr = (inst >> 9) & 0x7;
    int sr1 = (inst >> 6) & 0x7;
    if (flag) {
        // and inst number
        int16_t imm = SignExtend<int16_t, 5>(inst & 0b11111);
        reg[dr] = reg[sr1] & imm;
    } else {
        // add register
        int sr2 = inst & 0x7;
        reg[dr] = reg[sr1] & reg[sr2];
    }
    // Update condition register
    UpdateCondRegister(dr);
}

```

其余函数均类似。

## 四、编译

使用cmake -Bbuild, make指令进行编译。输入相关参数显示结果。

## 五、代码截图

```
0010001000010101
0100100000001000
0101010001100111
0001001010000100
0001000001111001
000000111111011
0001000001111001
0000100000000001
0001001001111001
1111000000100101
0101010010100000
0101011011100000
0101100100100000
0001010010100001
0001011011101000
0101101011000001
0000010000000001
0001100010000100
0001010010000010
0001011011000011
000010111111010
R0 = 10, R1 = 0, R2 = 2000, R3 = 0
R4 = 0, R5 = 0, R6 = 0, R7 = 3002
COND[NZP] = 010
PC = 3016

cycle = 49
请输出要打印的值的内存地址
|
```

上面为输入的代码片段，下面为运行完机器的状态，还有输出任何地址的值的功能。

## 六、总结与感悟

经过这次实验，提升了自己读代码和写代码能力，逐步接触了C++，以及它的一些特性。